CLASSIFIEZ AUTOMATIQUEMENT DES BIENS DE CONSOMMATION

Presentation - 25/05/24

place de marché

Rappel de la problématique

L’objectif principal de ce projet est d'étudier la **faisabilité de créer un moteur de classification** en utilisant à la fois des données **textuelles et des images**. Nous avons exploré différentes techniques de traitement du **langage naturel (NLP) et de traitement d'images**, en partant des méthodes basiques pour aller vers des techniques plus avancées.

Vision

L'idée est de déterminer quelles méthodes offrent les **meilleures performances pour classifier les articles de manière automatique**.

Présentation du jeu de données

Données textuelles

Les textes proviennent des descriptions produit. Chaque description a été prétraité pour enlever les éléments non pertinents, puis différentes techniques de **vectorisation** et d'**encodage** ont été appliquées. Le volume total des données textuelles s'élève à **1050 descriptions,** représentant un total de **80 013 mots.**

Données d'images

Les images sont associées aux produits et servent à enrichir la classification. Les images ont été traitées pour **améliorer leur qualité** et **extraire des features significatives**. Nous avons un total de **1050 images**.

**Les données proviennent de sources fiables et respectent les normes de propriété intellectuelle et de confidentialité des données.*

['uniq_id', 'product_name', 'image', 'description', 'product_category_tree', 'product_category', 'main_category', 'sub_category', 'description_clean', 'label_main_category']

Sommaire

01 Prétraitement des données textes

02 Méthodes NLP basiques d’encodage de texte

03 Méthodes NLP avancées d’encodage de texte

04 Méthodes de traitement d’image basique (SIFT)

05 Méthodes de traitement d’image plus avancée (CNN Transfer Learning)

06 classification supervisée d'images via CNN Transfer Learning

01. PRÉTRAITEMENT DES DONNÉES TEXTE

Prétraitements des textes

Nettoyage des données textes

- Traitement des valeurs manquantes
- Extraction des niveaux de catégories

product_category_tree	main_category
Home Decor & Festive Needs >> Showpieces	Home Decor & Festive Needs

- Traitement de la colonne “description” (Suppression des caractères non alphabétiques et conversion en minuscules)

	description	description_clean
0	Key Features of Elegance Polyester Multicolor ...	key features elegance polyester multicolor abs...
1	Specifications of Sathiyas Cotton Bath Towel (...	specifications sathiyas cotton bath towel bath...
2	Key Features of Eurospa Cotton Terry Face Towe...	key features eurospa cotton terry face towel s...
3	Key Features of SANTOSH ROYAL FASHION Cotton P...	key features santosh royal fashion cotton prin...
4	Key Features of Jaipur Print Cotton Floral Kin...	key features jaipur print cotton floral king s...

Méthodes de normalisation des mots

- **Stemming** : Le stemming réduit les mots à leur racine commune.
- **Lemmatization** : La lemmatisation est plus structurée que le stemming et vise à ramener le mot à sa forme de base correcte en tenant compte de son contexte.

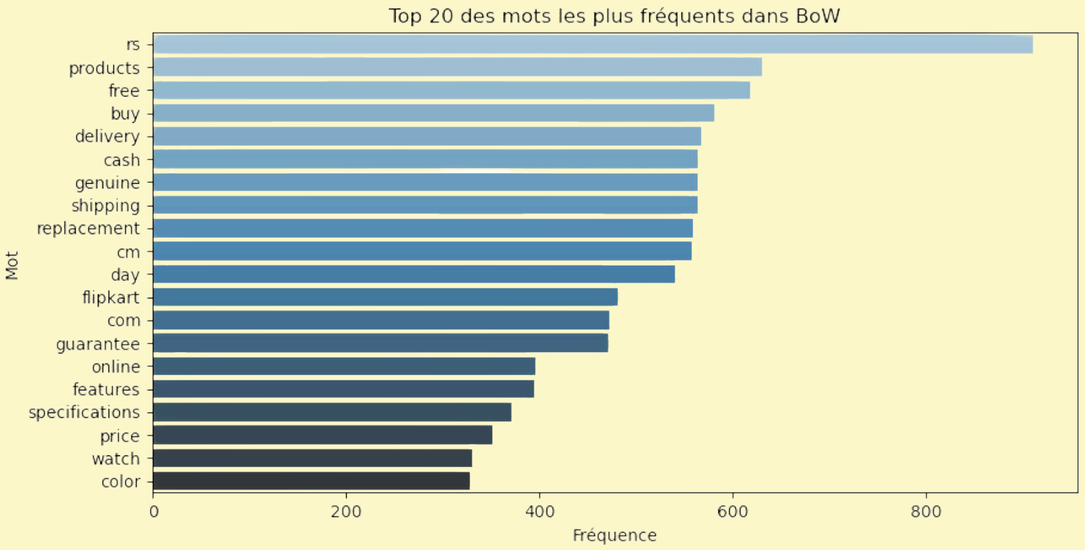
	description_clean	description_stemmed
1	specifications sathiyas cotton bath towel bath towel red yellow blue bath towel features machine washable yes material cotton design self design general brand sathiyas type bath towel gsm model name sathiyas cotton bath towel ideal men women boys girls model id asvtwl color red yellow blue size mediam dimensions length inch width inch box number contents sales package sales package bath towel	specif sathiya cotton bath towel bath towel red yellow blue bath towel featur machin washabl ye materi cotton design self design gener brand sathiya type bath towel gsm model name sathiya cotton bath towel ideal men women boy girl model id asvtwl color red yellow blue size mediam dimens length inch width inch box number content sale packag sale packag bath towel

description_tokenized	description_lemmatized
[specifications, sathiyas, cotton, bath, towel, bath, towel, red, yellow, blue, bath, towel, features, machine, washable, yes, material, cotton, design, self, design, general, brand, sathiyas, type. bath, towel, gsm, model, name, sathiyas, cotton, bath, towel, ideal, men, women, boys, girls, model, id, color, red, yellow, blue, size, mediam, dimensions, length, inch, width, inch, box, numbert contents, sales, package, sales, package, bath, towel]	specification sathiyas cotton bath towel bath towel red yellow blue bath towel feature machine washable yes material cotton design self design general brand sathiyas type bath towel gsm model name sathiyas cotton bath towel ideal men woman boy girl model id asvtwl color red yellow blue size mediam dimension length inch width inch box number content sale package sale package bath towel

02. MÉTHODES NLP BASIQUES D’ENCODAGE DE TEXTE

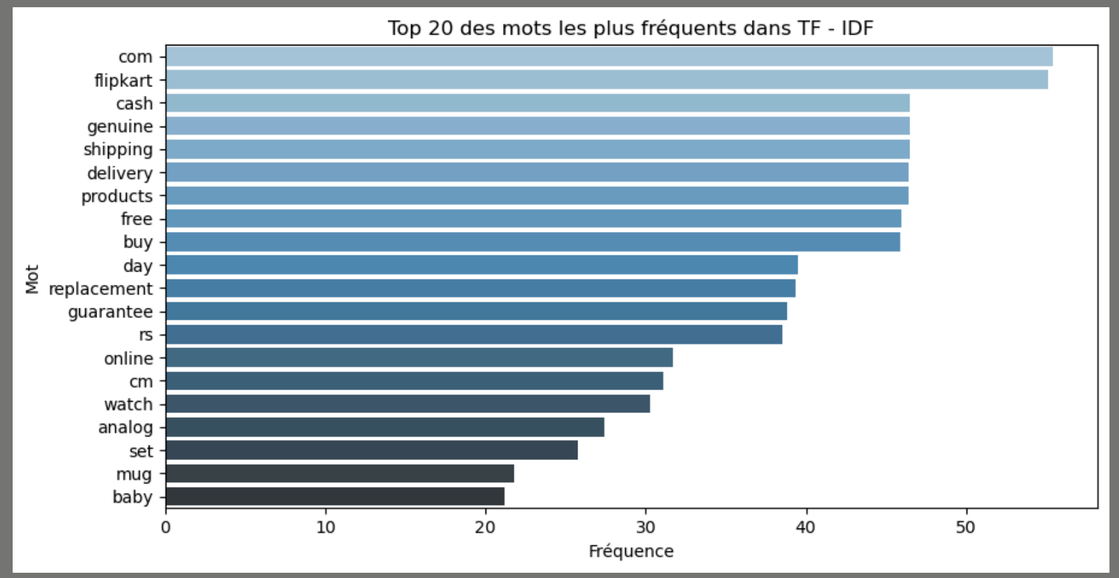
Encodage Bag of Words (BoW)

Méthode pour représenter les textes en **comptant combien de fois chaque mot apparaît. On ignore l'ordre des mots et la grammaire,** et on crée une liste de tous les mots uniques dans le texte, puis on compte le nombre de fois que chaque mot apparaît.



TF - IDF

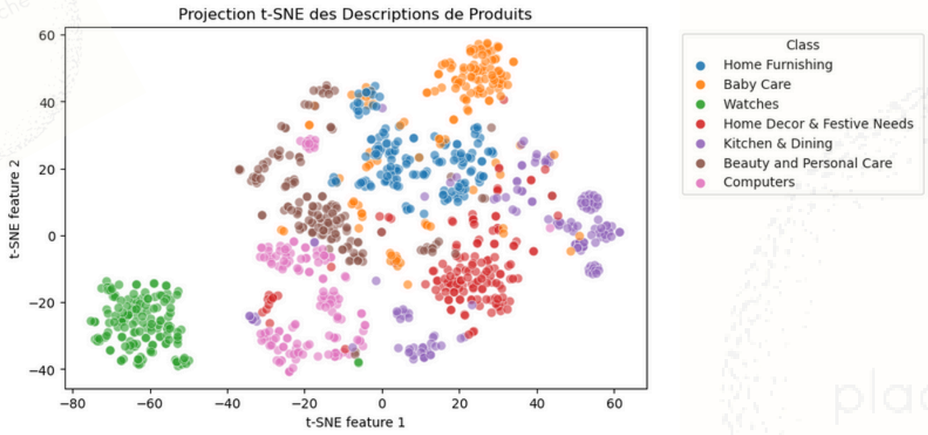
Méthode pour **évaluer l'importance des mots dans un texte.** Elle combine deux choses : **la fréquence du terme (TF)**, qui compte combien de fois un mot apparaît dans un document, **et la fréquence inverse du document (IDF)**, qui **réduit l'importance des mots courants** dans tous les documents.

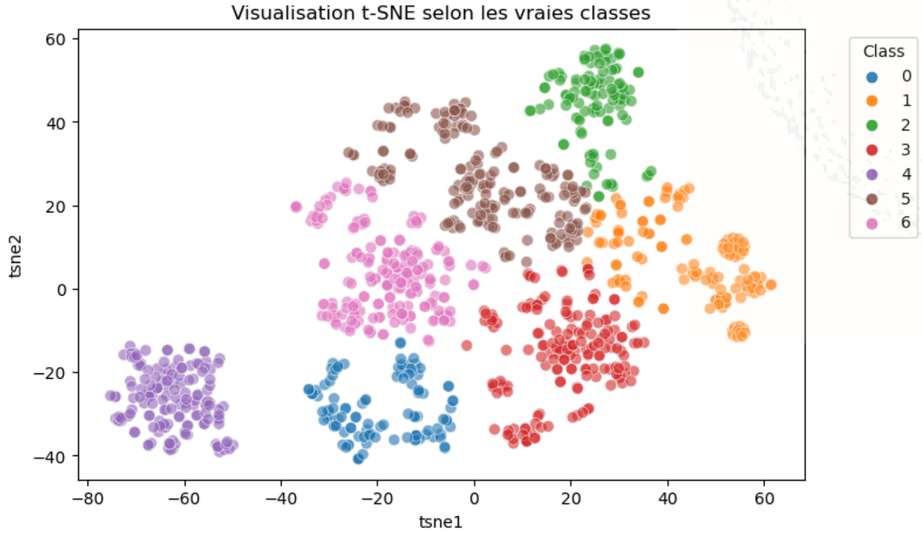


Implications pour la Classification :

TF-IDF est plus adapté pour identifier des termes distinctifs et pertinents pour la classification des documents.

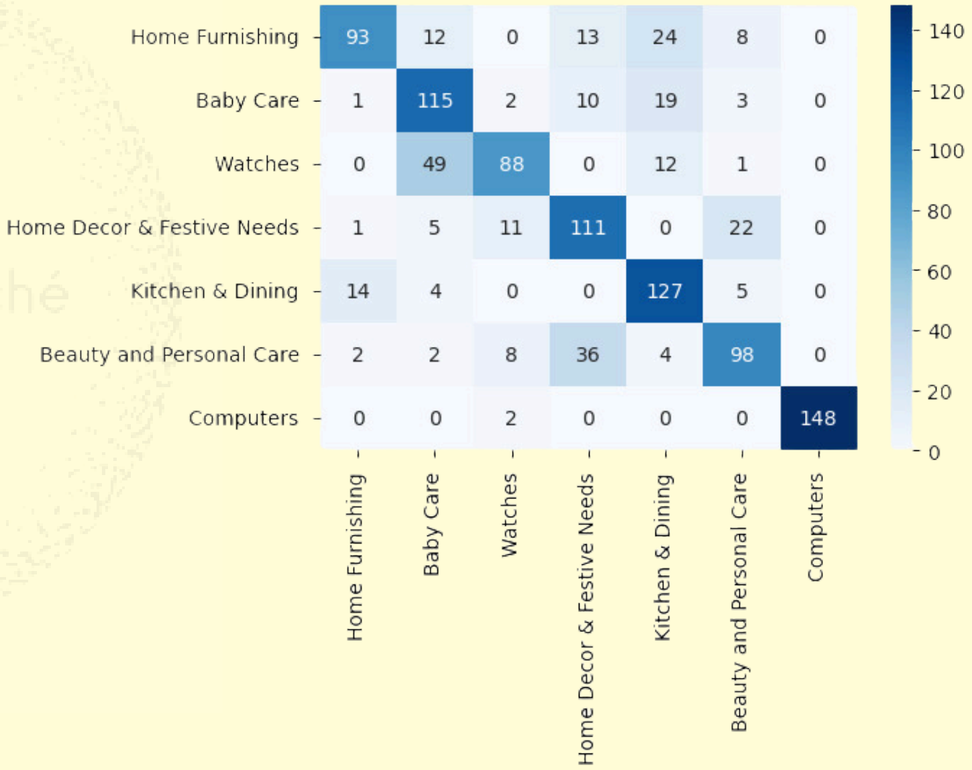
Réduction de dimension sur le texte encodé (TF-IDF)





ARI : 0.5249415037976954

Analyse par classes



03. MÉTHODES NLP AVANCÉES D’ENCODAGE DE TEXTE

Embeddings utilisés

Les premiers 5 valeurs de chaque embedding montrent une variation, ce qui indique que chaque description est représentée de manière unique par le modèle USE. Cela signifie que le modèle est capable de distinguer efficacement entre les différentes descriptions basées sur leur contenu sémantique.

Word2Vec

Les mots les plus similaires à 'cotton' sont :

- **'shorts':** 0.9977610111236572,
- **'care':** 0.9976004362106323
- **'care':** 0.9976004362106323
- **'fit':** 0.9966264963150024
- **'machine':** 0.9958372116088867,
- **'brand':** 0.9957334399223328,
- **'lining':** 0.9956885576248169,
- **'cotonex':** 0.9950529932975769,
- **'suit':** 0.9950205683708191,
- **'full':** 0.9950119853019714,
- **'model':** 0.9942931532859802

BERT

Phrases à analyser :

- text1 = "100% cotton dress"
- text2 = "synthetic fiber garment"
- text3 = "shirt color red"
- text4 = "Washing type 30 degree"

Similarité entre **'100% cotton dress'** et 'synthetic fiber garment': **0.7636**

Similarité entre **'100% cotton dress'** et **'shirt color red'**: 0.6876

Similarité entre **'100% cotton dress'** et **'Washing type 30 degree'**: 0.7288

Universal Sentence Encoder

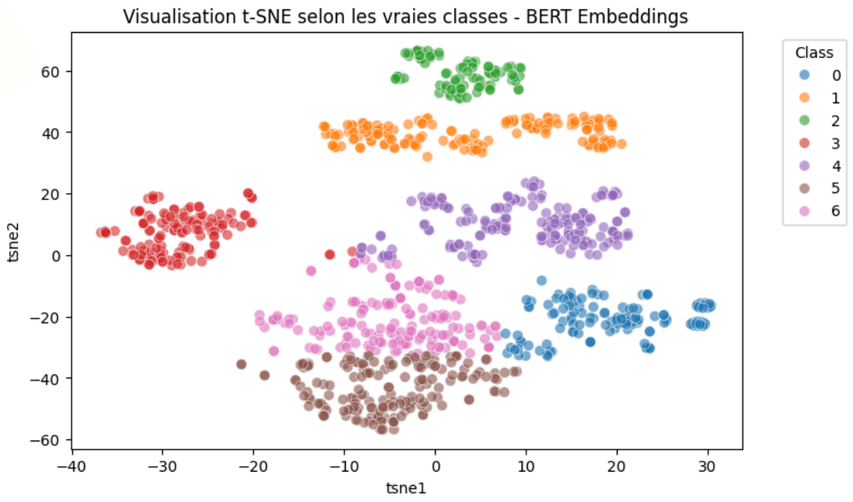
Descriptions : "Flipkart.com: Buy Engage Rush , Jump Combo Set for Rs. from Flipkart.com. - Lowest Prices, Only Genuine Products, 30 Day Replacement Guarantee, Free Shipping. Cash On Delivery!"

Résultat :

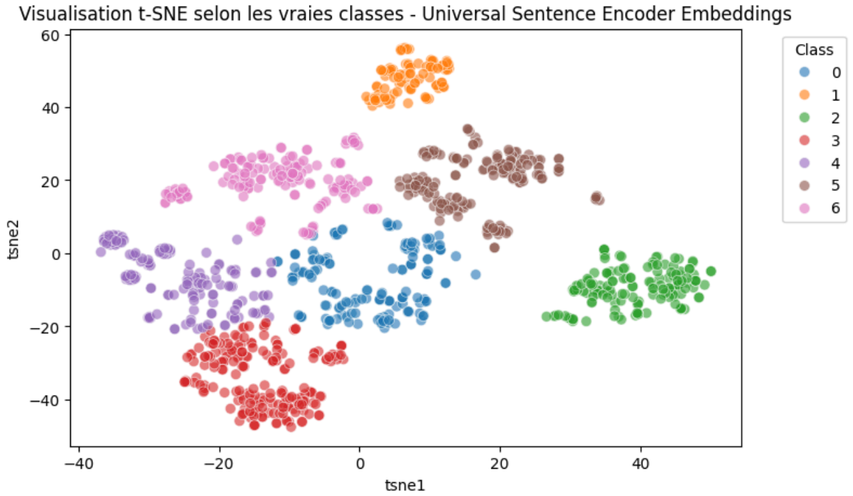
Embedding 1 :
- **Shape** : (512,)
- **Premiers 5 Valeurs** : [-0.01812908, -0.07984424, 0.00997086, 0.00629711, 0.02184289]

Embedding 2 :
- **Shape** : (512,)
- **Premiers 5 Valeurs** : [-0.05593703, -0.05931723, -0.01530571, 0.01872221, 0.04257472]

Visualisation des Clusters par t-SNE pour Différentes Méthodes d'Embedding

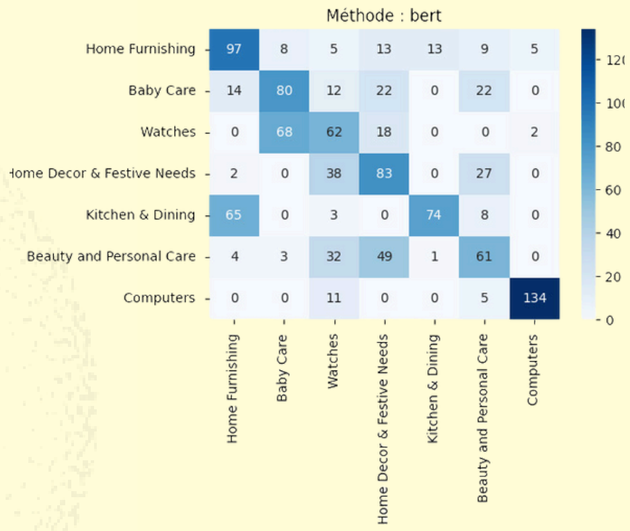


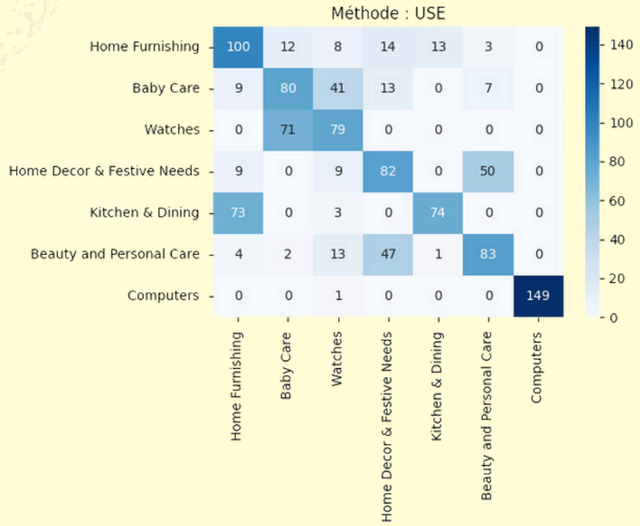
ARI : 0.34213206161217685



ARI : 0.42347785427505674

Analyse par classes





04. MÉTHODES DE TRAITEMENT D’IMAGE BASIQUE (SIFT)

Préparation du dataset

data_clean

- 0 uniq_id
- 1 product_name
- 2 image
- 3 description
- 4 product_category_tree
- 5 main_category
- 6 label_main_category
- 7 sub_category
- 8 description_clean

data

- 0 image
- 1 main_category
- 2 label_main_category

Le nombre total de descripteurs SIFT extraits de toutes les images est de 517 351. L'algorithme SIFT a trouvé 517 351 points d'intérêt à travers toutes les images.

Chaque descripteur SIFT a une dimension de 128. Cela correspond à la taille fixe des descripteurs SIFT, qui est toujours de 128 valeurs. Ces valeurs représentent des caractéristiques locales autour de chaque point d'intérêt détecté dans l'image.

SIFT

Pré-traitement des images via SIFT

Nombre de descripteurs :

517 351

Cluster

Création des clusters de descripteurs

Nombre de Cluster :

719

Features

Création des features des images

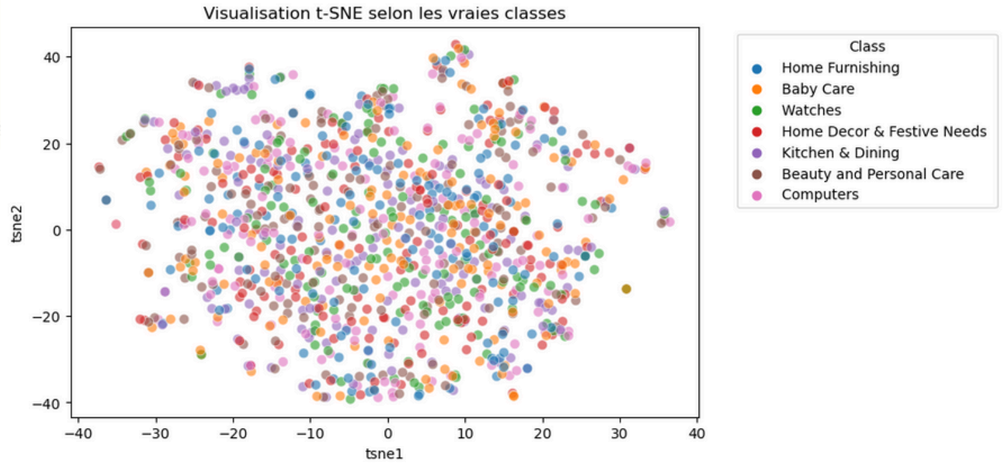
Pour chaque image :

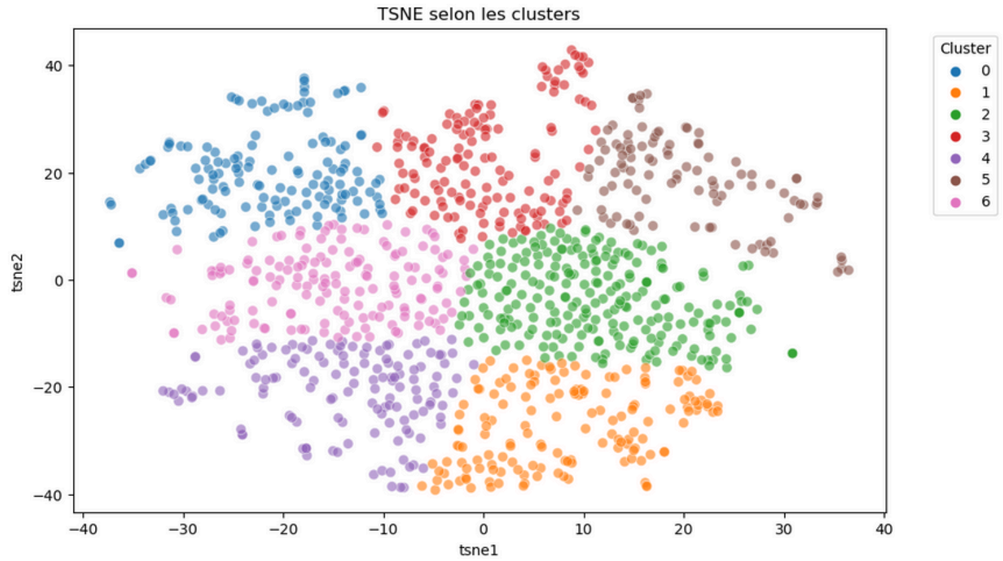
- Prédiction des numéros de cluster de chaque descripteur
- Création d'un histogramme comptant pour chaque numéro de cluster le nombre de descripteurs de l'image
- Les features d'une image sont représentées par l'histogramme, qui compte le nombre de descripteurs par cluster

Temps de création des histogrammes :

4.36 s

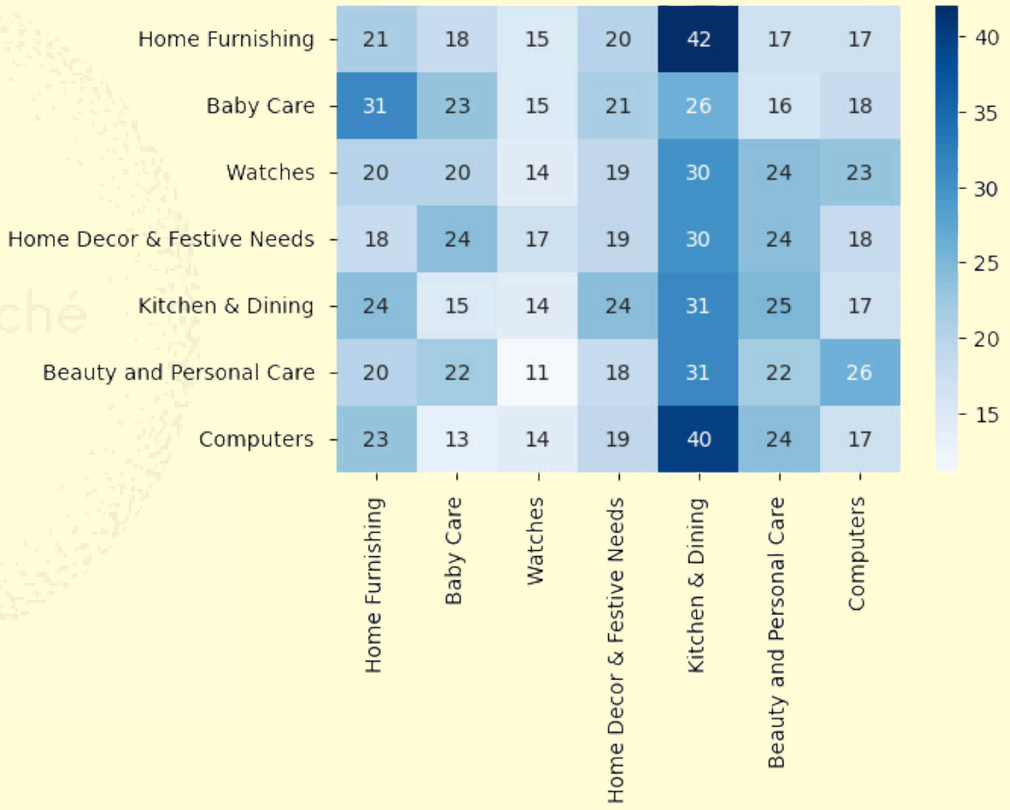
Visualisation des Clusters par t-SNE pour Différentes Méthodes d'Embedding





ARI : -0.0010452854002161933

Analyse par classes



05. MÉTHODES DE TRAITEMENT D’IMAGE AVANCÉES

Préparation des datasets

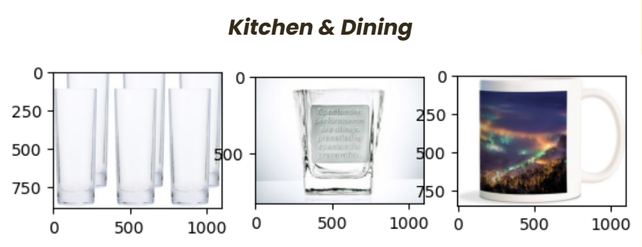
- data
  - dataset_test
    - Baby Care
    - Beauty and Personal Care
    - Computers
    - Home Decor & Festive Needs
    - Home Furnishing
    - Kitchen & Dining
    - Watches
  - dataset_train
    - Baby Care
    - Beauty and Personal Care
    - Computers
    - Home Decor & Festive Needs
    - Home Furnishing
    - Kitchen & Dining
    - Watches
  - Images

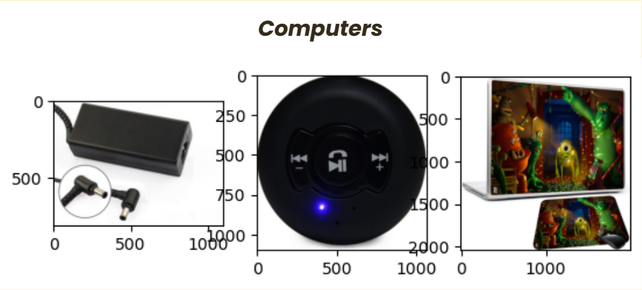
```
data.sample(5)
```

	image_path	label_name	label
581	./data/dataset_train\Home Furnishing\dc777afc9...	Home Furnishing	4
751	./data/dataset_train\Watches\4019ad332ca3ce224...	Watches	6
780	./data/dataset_train\Watches\7264f56639c8c5415...	Watches	6
743	./data/dataset_train\Watches\30305ade337a4dde1...	Watches	6
315	./data/dataset_train\Computers\9dc48a756ba86bb...	Computers	2

```
data_test.sample(5)
```

	image_path	label_name	label
52	./data/dataset_test\Computers\215decf6f993d7b4...	Computers	2
119	./data/dataset_test\Home Furnishing\159b45e4de...	Home Furnishing	4
95	./data/dataset_test\Home Decor & Festive Needs...	Home Decor & Festive Needs	3
33	./data/dataset_test\Beauty and Personal Care\2...	Beauty and Personal Care	1
75	./data/dataset_test\Computers\9f141b3c3d9096f5...	Computers	2





VGG16

VGG16 est un modèle de réseau de neurones convolutifs profond qui a été **entraîné sur l'ensemble de données ImageNet**

- Les images sont chargées et **redimensionnées à 224x224 pixels**, conformément à la taille d'entrée attendue par VGG16.
- Les images sont **converties en tableau numpy** et prétraitées pour correspondre aux attentes du modèle VGG16.
- L'extraction des caractéristiques à partir de VGG16 permet de convertir les images en vecteurs de caractéristiques de haute dimension (4096 dimensions) qui encapsulent les informations pertinentes pour la classification.

Model: "functional_1"

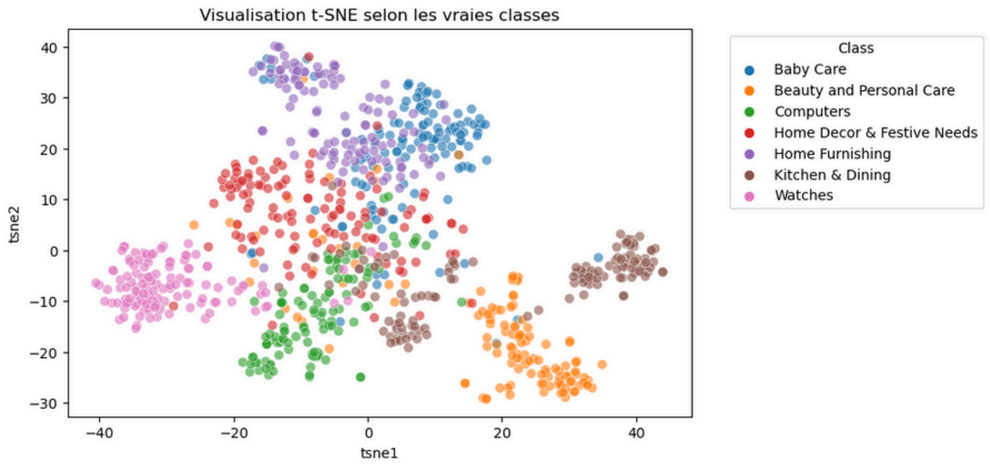
Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102,764,544
fc2 (Dense)	(None, 4096)	16,781,312

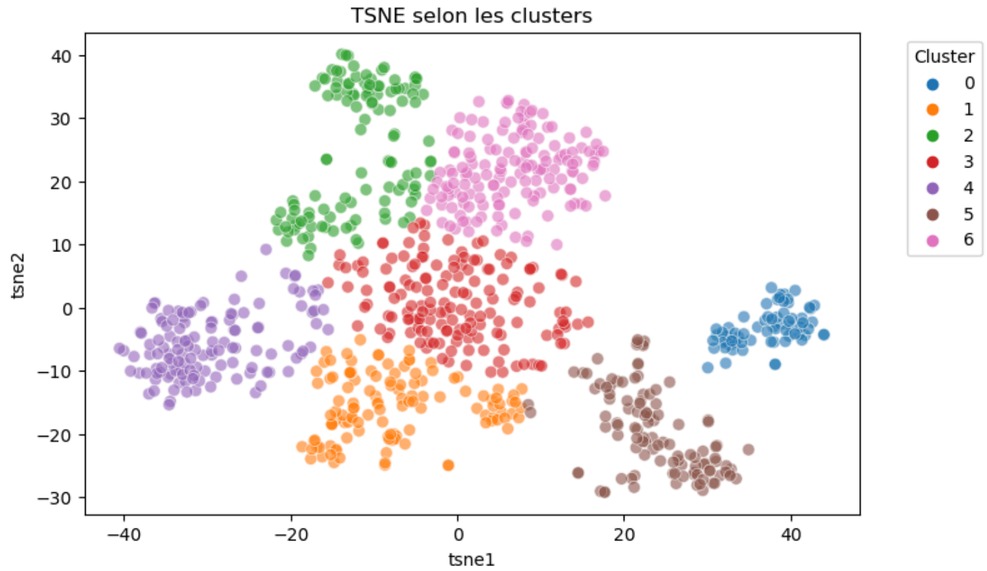
Total params: 134,260,544 (512.16 MB)

Trainable params: 134,260,544 (512.16 MB)

Non-trainable params: 0 (0.00 B)

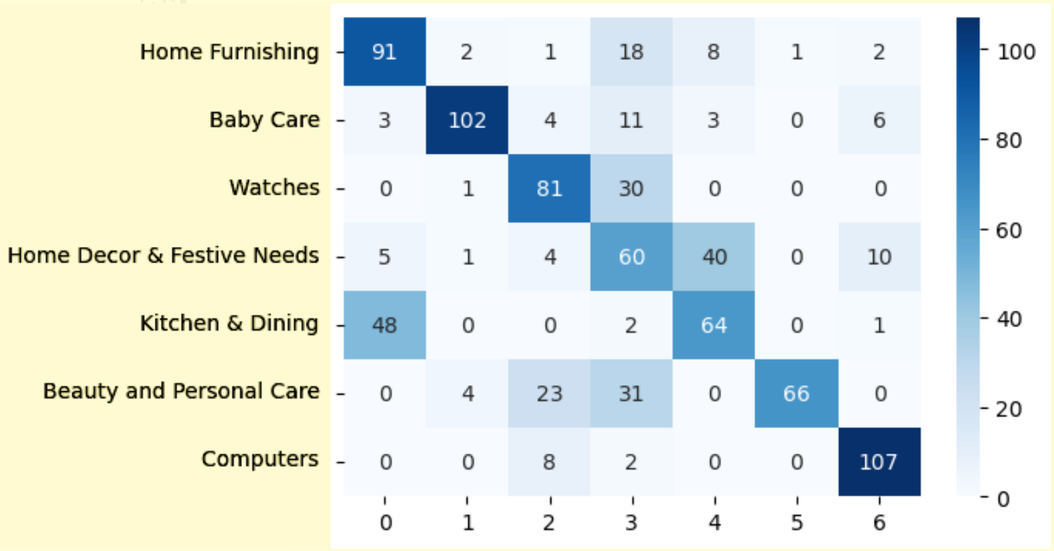
Visualisation des Clusters par t-SNE pour Différentes Méthodes d'Embedding





ARI : 0.45808472479249435

Analyse par classes



VGG19

VGG19 est une **variante plus profonde de VGG16**, comportant **plus de couches convolutives** et de **paramètres**, ce qui peut potentiellement améliorer la capacité du modèle à extraire des caractéristiques pertinentes des images.

Model: "functional_3"

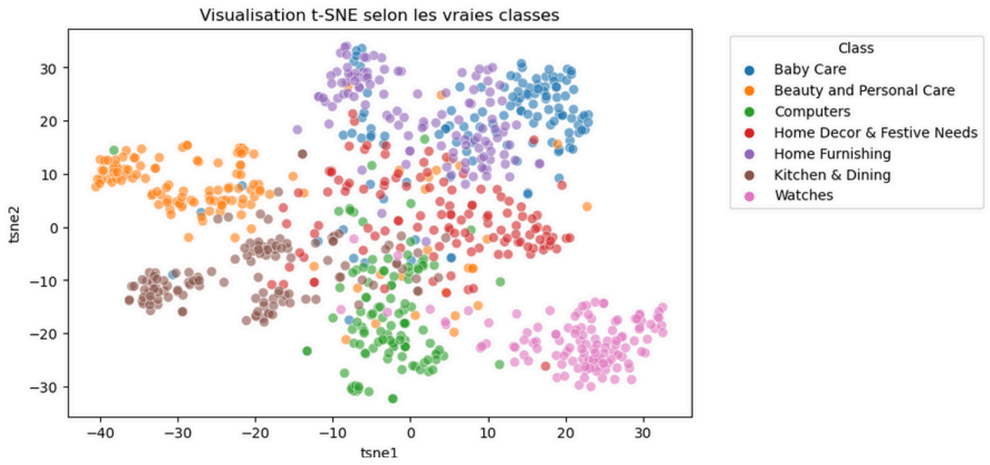
Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102,764,544
fc2 (Dense)	(None, 4096)	16,781,312

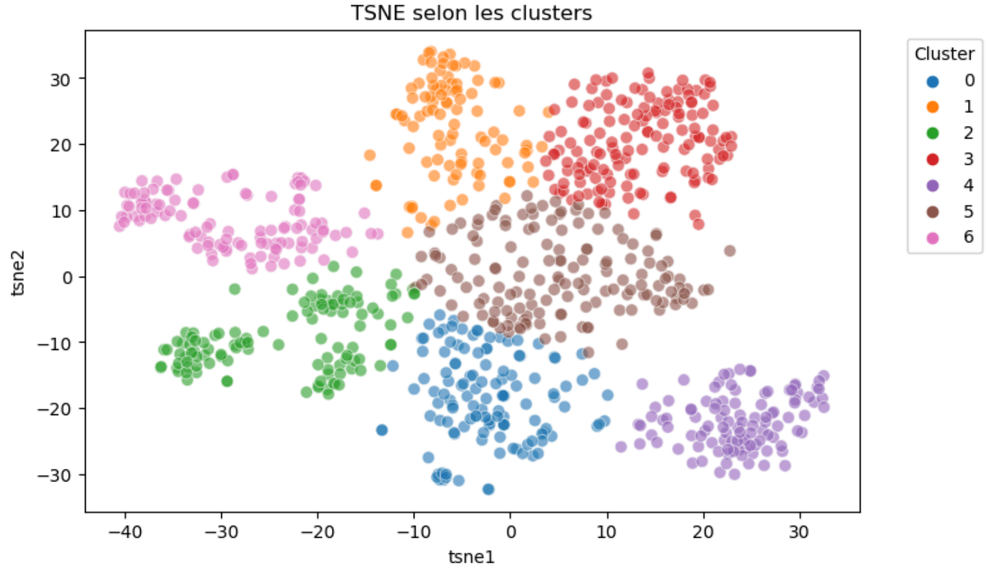
Total params: 139,570,240 (532.42 MB)

Trainable params: 139,570,240 (532.42 MB)

Non-trainable params: 0 (0.00 B)

Visualisation des Clusters par t-SNE pour Différentes Méthodes d'Embedding

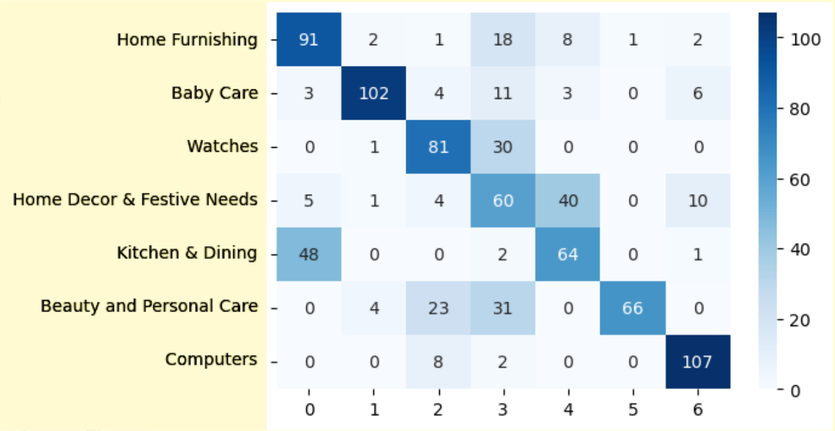




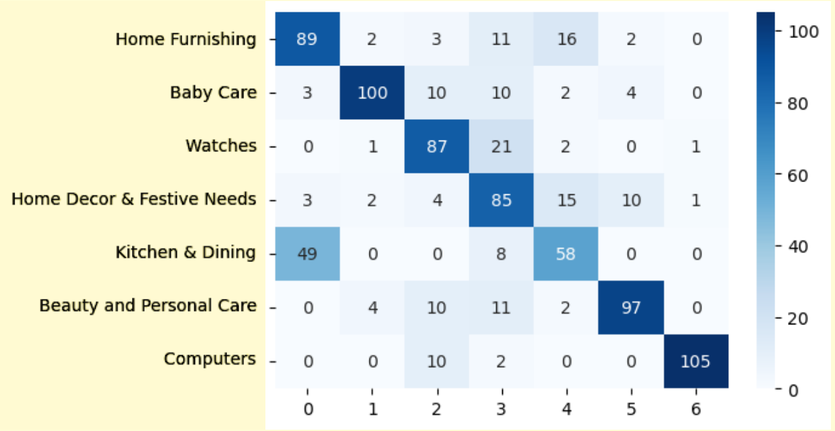
ARI : 0.5233273312403294

Analyse par classes

VGG16



VGG19



06. CLASSIFICATION SUPERVISÉE D'IMAGES VIA CNN TRANSFER LEARNING

place de marché

Préparation Initiale des Images

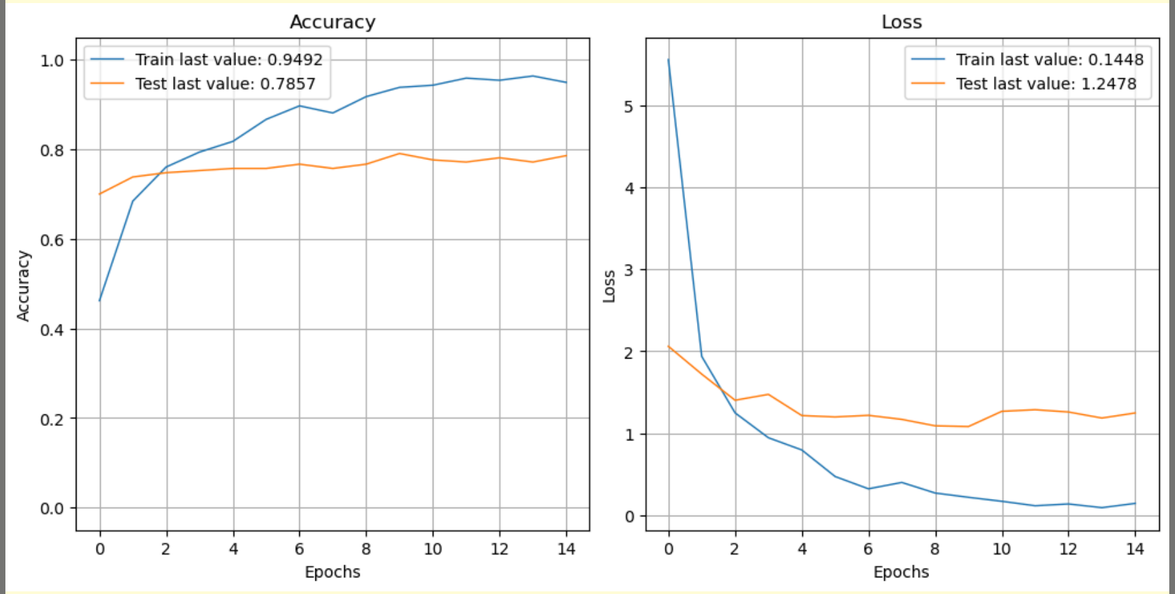
Procésus :

- **Redimensionnement des images :** images redimensionnées à **224x224 pixels**
- **Conversion en tableaux numpy :** Les images redimensionnées ont été converties en tableaux numpy **(matrices)** pour être traitées par les modèles de deep learning.
- **Prétraitement des images :** Les images ont été prétraitées en utilisant la fonction `**preprocess_input**` de Keras, qui normalise les pixels des images dans le même format que celui utilisé lors de l'entraînement initial du modèle VGG sur ImageNet.
- **Création des tensors d'entraînement et de test :** Les tensors d'images préparées ont été divisés en ensembles d'entraînement et de test pour permettre la validation croisée et l'évaluation des modèles.

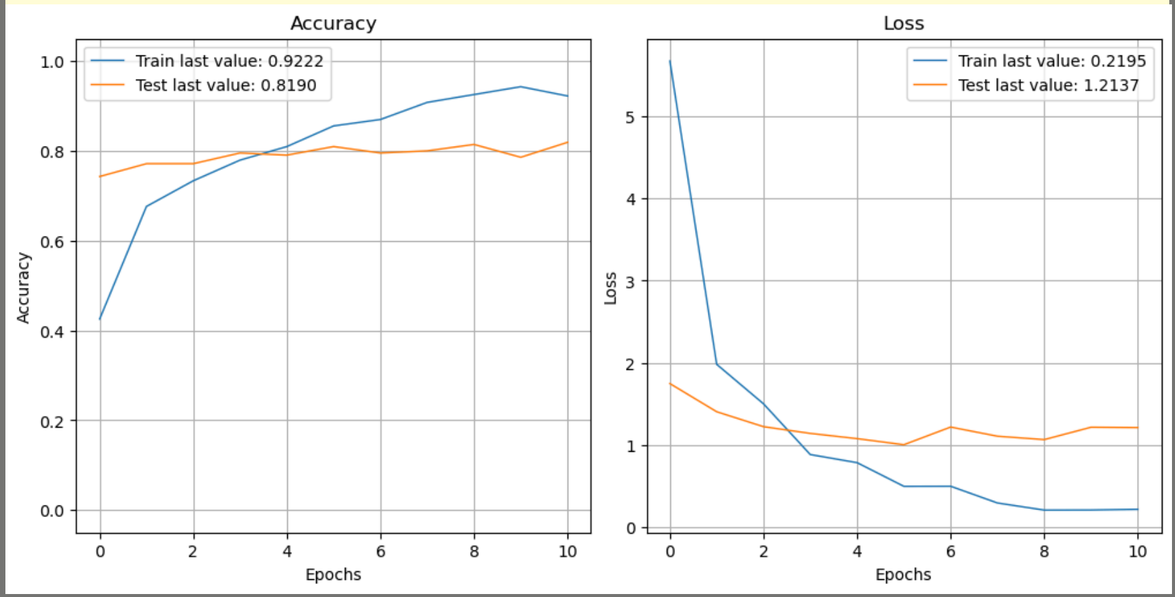
Observations VGG19 :

- *Meilleure accuracy de test (0.8190 contre 0.7857)*
- *Perte de test légèrement inférieure (1.2137 contre 1.2478)*
- *Moins d'overfitting, suggéré par un écart moindre entre les performances d'entraînement et de test.*

VGG16



VGG19



SCORE

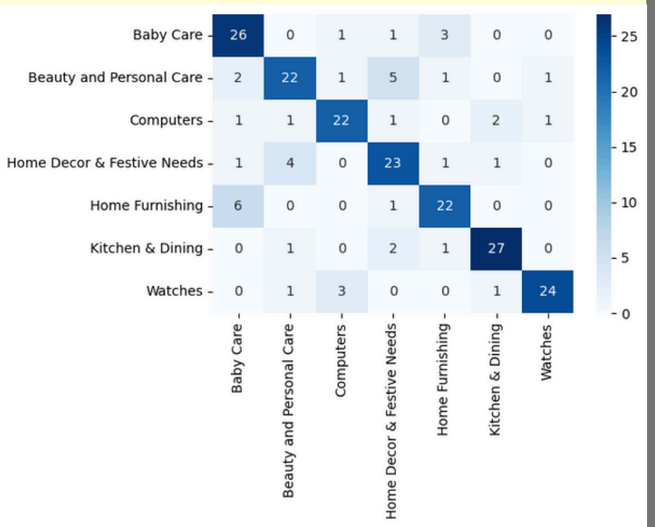
VGG16

	precision	recall	f1-score	support
0	0.72	0.84	0.78	31
1	0.76	0.69	0.72	32
2	0.81	0.79	0.80	28
3	0.70	0.77	0.73	30
4	0.79	0.76	0.77	29
5	0.87	0.87	0.87	31
6	0.92	0.83	0.87	29
accuracy			0.79	210
macro avg	0.80	0.79	0.79	210
weighted avg	0.79	0.79	0.79	210

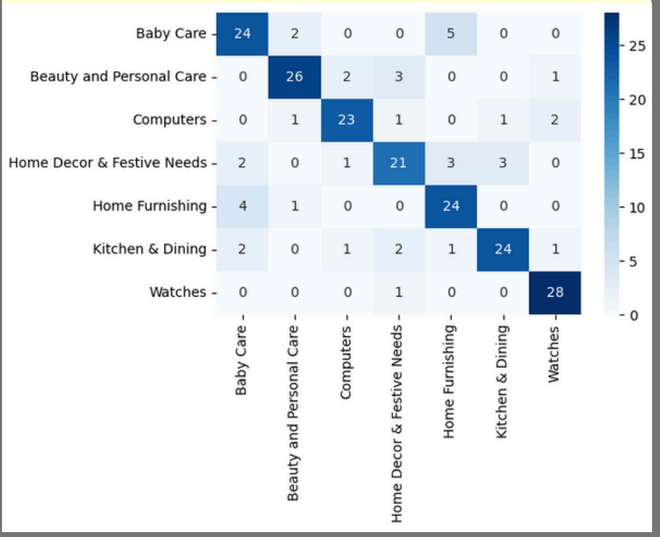
VGG19

	precision	recall	f1-score	support
0	0.75	0.77	0.76	31
1	0.87	0.81	0.84	32
2	0.85	0.82	0.84	28
3	0.75	0.70	0.72	30
4	0.73	0.83	0.77	29
5	0.86	0.77	0.81	31
6	0.88	0.97	0.92	29
accuracy			0.81	210
macro avg	0.81	0.81	0.81	210
weighted avg	0.81	0.81	0.81	210

VGG16



VGG19



Approche Nouvelle par Dataset avec Data Augmentation Intégrée au Modèle

Processus:

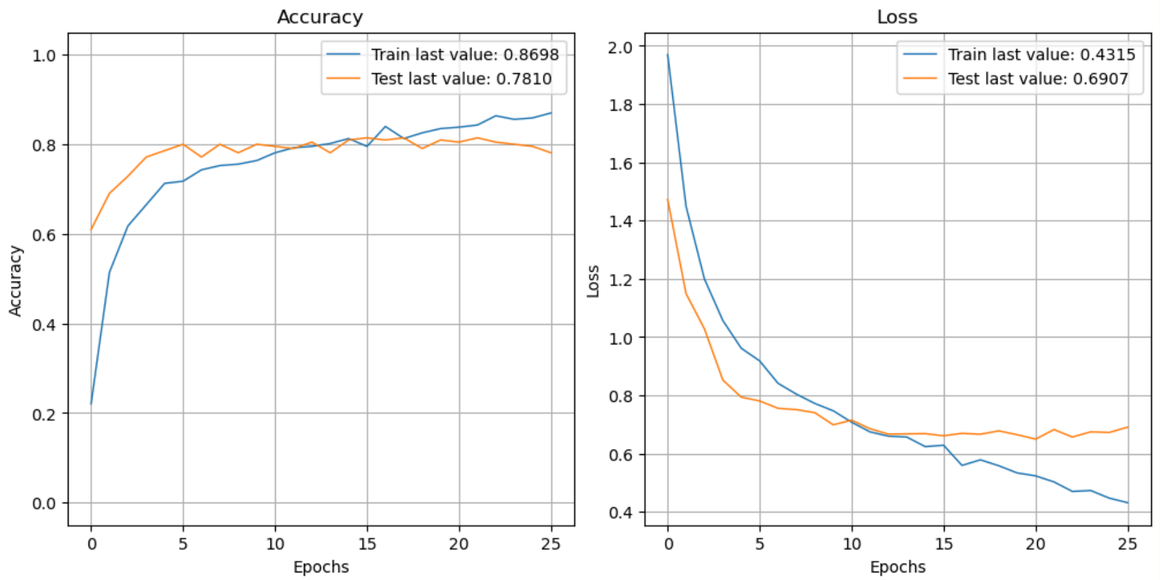
- Dans cette approche, les images sont chargées en lots pendant l'entraînement avec des **transformations aléatoires appliquées** à la volée (**data augmentation**).

- Utilisation d'une **pipeline de traitement** des données pour appliquer des augmentations comme des **rotations, des flips, des zooms, etc., directement pendant l'entraînement.**

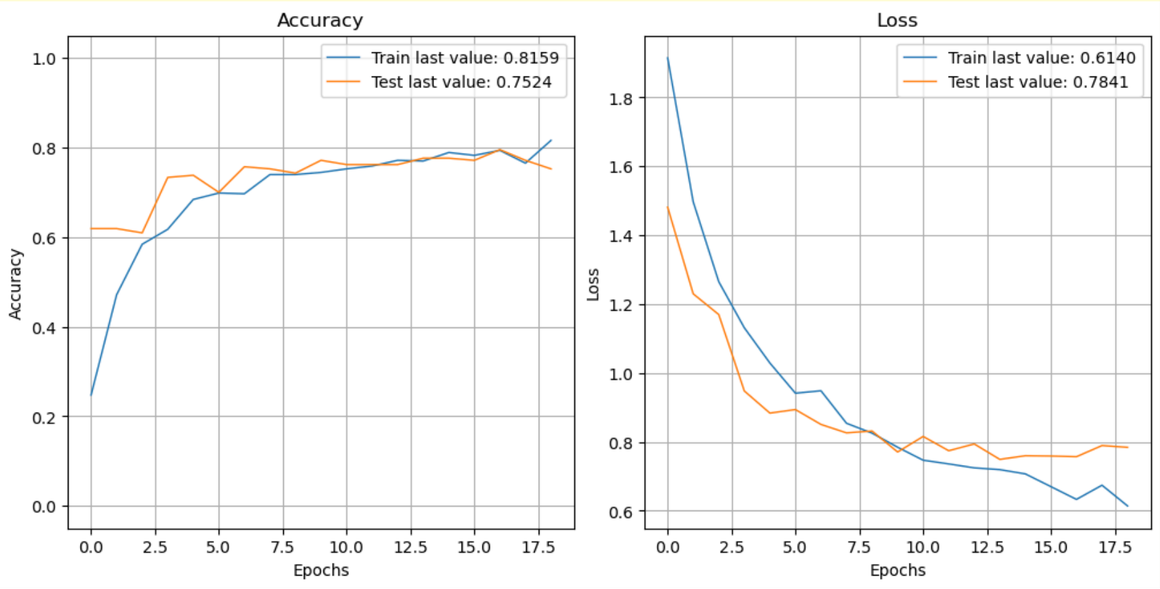
Inconvénients:

- **Complexité:**
  - Nécessite une **configuration plus complexe pour la pipeline** de traitement des données.
  - Peut **augmenter le temps de calcul pendant l'entraînement** en raison des transformations appliquées à la volée.

VGG16



VGG19



SCORE

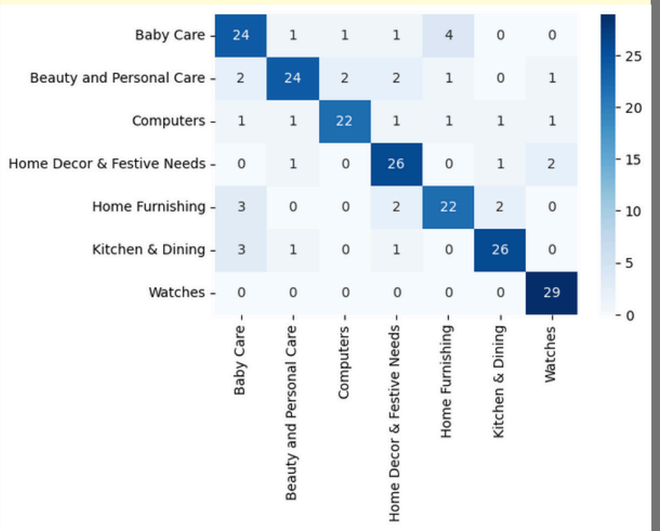
VGG16

	precision	recall	f1-score	support
0	0.73	0.77	0.75	31
1	0.86	0.75	0.80	32
2	0.88	0.79	0.83	28
3	0.79	0.87	0.83	30
4	0.79	0.76	0.77	29
5	0.87	0.84	0.85	31
6	0.88	1.00	0.94	29
accuracy			0.82	210
macro avg	0.83	0.82	0.82	210
weighted avg	0.83	0.82	0.82	210

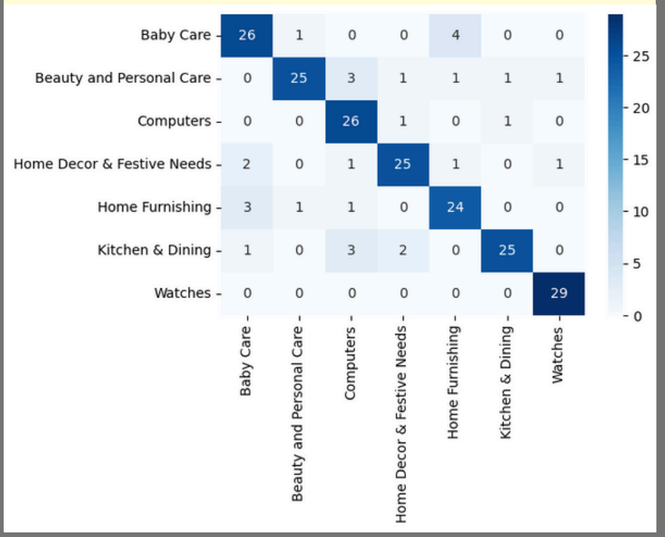
VGG19

	precision	recall	f1-score	support
0	0.81	0.84	0.83	31
1	0.93	0.78	0.85	32
2	0.76	0.93	0.84	28
3	0.86	0.83	0.85	30
4	0.80	0.83	0.81	29
5	0.93	0.81	0.86	31
6	0.94	1.00	0.97	29
accuracy			0.86	210
macro avg	0.86	0.86	0.86	210
weighted avg	0.86	0.86	0.86	210

VGG16



VGG19



Model	Approach	Training Accuracy	Validation Accuracy	Test Accuracy	Validation Loss	Test Loss	Precision, Recall, F1-Score
VGG16	Préparation Initiale des Images	99.84	79.05	80.0	1.08222	1.2478	0.79
VGG19	Préparation Initiale des Images	99.52	80.95	77.62	1.00531	1.2137	0.81
VGG16	Dataset avec Data Augmentation	90.0	81.05	80.48	0.66106	0.8048	0.82
VGG19	Dataset avec Data Augmentation	86.51	77.62	75.71	0.74919	0.7571	0.86

Conclusion :

Le modèle VGG16 avec Dataset et Data Augmentation intégrée au modèle *est recommandé en raison de sa meilleure capacité de généralisation, de ses bonnes performances sur les ensembles de validation et de test, et de ses métriques de classification élevées. Cela montre que ce modèle est* ***moins susceptible de surajuster*** *et est* ***plus robuste pour des prédictions précises sur des données réelles.***

```python
def fetch_champagne_data(ingredient, num_products):
    url = "https://edamam-food-and-grocery-database.p.rapidapi.com/api/food-database/v2/parser"
    querystring = {
        "nutrition-type": "cooking",
        "category[0]": "generic-foods",
        "health[0]": "alcohol-free",
        "ingr": ingredient
    }
    headers = {
        "X-RapidAPI-Key": "[illegible]",
        "X-RapidAPI-Host": "edamam-food-and-grocery-database.p.rapidapi.com"
    }

    response = requests.get(url, headers=headers, params=querystring)

    if response.status_code == 200:
        data = response.json()
        products = data['hints'][:num_products]
        product_list = []

        for product in products:
            food = product.get('food', {})
            product_info = {
                'foodId': food.get('foodId'),
                'label': food.get('label'),
                'category': food.get('category'),
                'foodContentsLabel': food.get('foodContentsLabel'),
                'image': food.get('image')
            }
            product_list.append(product_info)

        df = pd.DataFrame(product_list)
        return df
    else:
        raise Exception(f"Error: {response.status_code}, {response.text}")

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/fetch-data', methods=['POST'])
def fetch_data():
    ingredient = request.form.get('ingredient', 'champagne')
    num_products = int(request.form.get('num_products', 10))
    try:
        df = fetch_champagne_data(ingredient, num_products)
        return render_template('data.html', tables=[df.to_html(classes='data', header="true", index=False)], ingredient=ingredient, n
    except Exception as e:
        flash(str(e), 'error')
        return redirect(url_for('home'))

@app.route('/download-csv')
def download_csv():
    ingredient = request.args.get('ingredient', 'champagne')
    num_products = int(request.args.get('num_products', 10))
    try:
        df = fetch_champagne_data(ingredient, num_products)
        str_io = BytesIO()
        df.to_csv(str_io, index=False)
        str_io.seek(0)
        return send_file(str_io, mimetype='text/csv', attachment_filename='champagne_products.csv', as_attachment=True)
    except Exception as e:
        flash(str(e), 'error')
```

127.0.0.1:5000

Récupérer et afficher les données à base d'un ingrédient sur Edamam Food and Grocery Database

Ingrédient: champagne Nombre de produits: 10 Fetch Data

Télécharger CSV

Merci

Pour votre attention